AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the above-identified application:

Claim 1 (Currently amended). A method for transferring CPU budget and CPU control between ~~and~~ a client thread and a server thread in a client/server pair, comprising:

assigning a CPU budget to said client thread, the CPU budget occurring within a first period;

executing said client thread at a scheduled time within [[a]] said first period;

transferring, within said first period, CPU control and any unused CPU budget to said server thread when said client thread stops executing;

executing said server thread within said first period; and

transferring, within said first period, CPU control and any unused CPU budget to said client thread when said server thread stops executing.

Claim 2 (Original). A method according to claim 1 further comprising alternately transferring CPU control and unused CPU budget between said client thread and said server thread within said first period.

Claim 3 (Original). A method according to claim 2 further comprising terminating the execution of said client thread and said server thread when said CPU budget has expired.

Claim 4 (Original). A method according to claim 3 wherein the first step of executing comprises transferring service requests from the client to the server.

Claim 5 (Original). A method according to claim 4 wherein the second step of executing comprises transferring results of the service requests from the server to the client.

Claim 6 (Original). A method according to claim 5 wherein said client thread places service request in a client-to-server queue when said client thread is executing and

wherein said server thread retrieves and processes the service request when said server thread is executing.

Claim 7 (Original). A method according to claim 6 wherein said server thread places the results of the service request in a server-to-client queue when the server thread is executing and wherein said client thread retrieves the results when said client thread is executing.

Claim 8 (Original). A method according to claim 7 wherein the first step of transferring occurs when said client thread has completed transferring service requests to said client-to-server queue.

Claim 9 (Original). A method according to claim 7 wherein the first step transferring occurs when said client-to-server queue is full.

Claim 10 (Original). A method according to claim 7 wherein the first step of transferring occurs when a service request must be processed immediately.

Claim 11 (Original). A method according to claim 7 wherein the second step of transferring occurs when said server-to-client queue is full.

Claim 12 (Original). A method according to claim 7 wherein the second step of transferring occurs when said server thread empties said client-to-server queue.

Claim 13 (Original). A method according to claim 7 wherein the second step of transferring occurs when said server thread is responding to a priority service request from said client thread.

Claim 14 (Original). A method according to claim 7 wherein the first step of transferring occurs upon the occurrence of a synchronization object.

Claim 15 (Original). A method according to claim 14 wherein the second step of transferring occurs upon the occurrence of a synchronization object.

Claim 16 (Original). A method according to claim 16 wherein said synchronization object is an event.

Claim 17 (Original). A method according to claim 15 wherein said synchronization object is a semaphore.

Claim 18 (Original). A method according to claim 1 wherein the CPU budget assigned to said client thread is sufficient to complete the task of the client/server pair.

Claim 19 (Original). A method according to claim 1 further comprising assigning a CPU budget to said server thread.

Claim 20 (Currently amended). A method for transferring CPU budget control between a client thread and a server thread in a client/server pair, comprising:

executing said client thread at a scheduled time within a first period;

transferring control of the CPU within said first period to said server thread when said client thread stops executing;

executing said server thread ~~in~~ within said first period; and

transferring within said first period, control of the CPU to said client thread when said server thread stops executing.

Claim 21 (Original). A method according to claim 20 further comprising alternately transferring CPU control between said client thread and said server thread within said first period.

Claim 22 (Original). A method according to claim 20 wherein the first step of executing comprises transferring service requests from the client to the server.

Claim 23 (Original). A method according to claim 22 wherein the second step of executing comprises transferring results of the service requests from the server to the client.

Claim 24 (Original). A method according to claim 23 wherein said client thread places service requests in a client-to-server queue when said client thread is executing and wherein said server thread retrieves and processes the service requests when said server thread is executing.

Claim 25 (Original). A method according to claim 24 wherein said server thread places the results of the service requests in a server-to-client queue when the server thread is executing and wherein said client thread retrieves the results when said client is executing.

Claim 26 (Original). A method according to claim 25 wherein the first step of transferring occurs when said client thread has completed transferring service requests to said client-to-server queue.

Claim 27 (Original). A method according to claim 25 wherein the first step of transferring occurs when said client-to-server queue is full.

Claim 28 (Original). A method according to claim 25 wherein the first step of transferring occurs when a service request must be processed immediately.

Claim 29 (Original). A method according to claim 25 wherein the second step of transferring occurs when said service to client queue is full.

Claim 30 (Original). A method according to claim 25 wherein the second step of transferring occurs when said server thread empties said client-to-server queue.

Claim 31 (Original). A method according to claim 25 wherein the second step of transferring occurs when said server thread is responding to a priority service request from said client thread.

Claim 32 (Original). A method according to claim 25 wherein the first step of transferring occurs upon the occurrence of a synchronization object.

Claim 33 (Original). A method according to claim 32 wherein the second step of transferring occurs upon the occurrence of a synchronization object.

Claim 34 (Original). A method according to claim 33 wherein said synchronization object is an event.

Claim 35 (Original). A method according to claim 33 wherein said synchronization object is a semaphore.